



FOR GOD AND COUNTRY

ST. LAWRENCE HIGH SCHOOL

A Jesuit Christian Minority Institution

STUDY MATERIAL - 11



Subject: COMPUTER SCIENCE

Class - 12

Chapter: Register and Counters

Date: 01/07/2020

Shift Register

A group of flip flops which is used to store multiple bits of data and the data is moved from one flip flop to another is known as **Shift Register**. The bits stored in registers shifted when the clock pulse is applied within and inside or outside the registers. To form an n-bit shift register, we have to connect n number of flip flops. So, the number of bits of the binary number is directly proportional to the number of flip flops. The flip flops are connected in such a way that the first flip flop's output becomes the input of the other flip flop.

A **Shift Register** can shift the bits either to the left or to the right. A **Shift Register**, which shifts the bit to the left, is known as "**Shift left register**", and it shifts the bit to the right, known as "**Right left register**".

The shift register is classified into the following types:

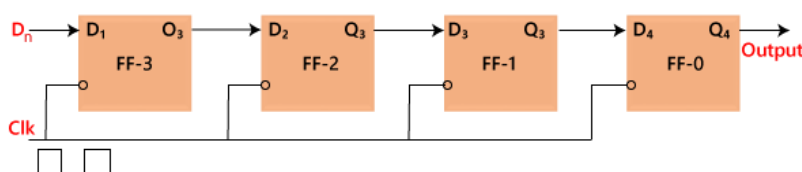
- Serial In Serial Out
- Serial In Parallel Out
- Parallel In Serial Out
- Parallel In Parallel Out

Serial IN Serial OUT

In "Serial Input Serial Output", the data is shifted "IN" or "OUT" serially. In SISO, a single bit is shifted at a time in either right or left direction under clock control.

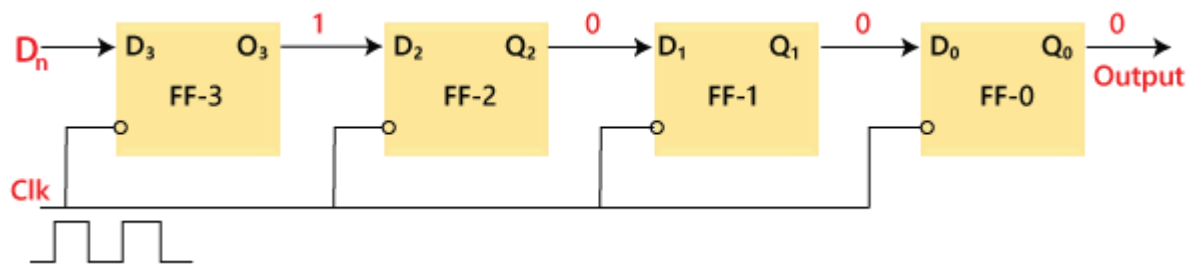
Initially, all the flip-flops are set in "reset" condition i.e. $Y_3 = Y_2 = Y_1 = Y_0 = 0$. If we pass the binary number 1111, the LSB bit of the number is applied first to the Din bit. The D3 input of the third flip flop, i.e., FF-3, is directly connected to the serial data input D₃. The output Y₃ is passed to the data input d₂ of the next flip flop. This process remains the same for the remaining flip flops. The block diagram of the "**Serial IN Serial OUT**" is given below.

Block Diagram:

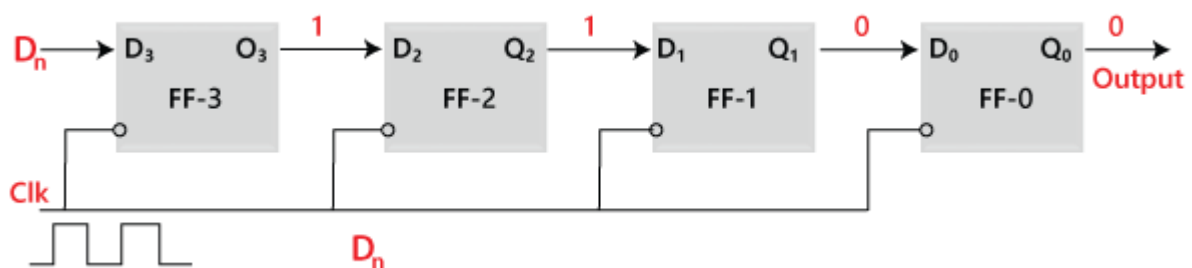


Operation

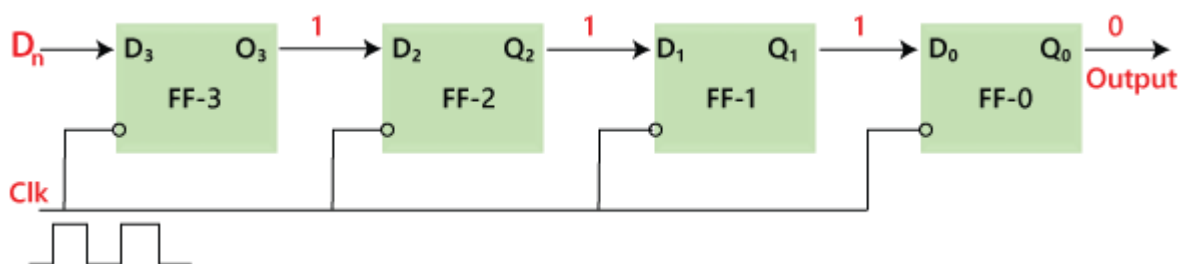
When the clock signal application is disabled, the outputs $Y_3 Y_2 Y_1 Y_0 = 0000$. The LSB bit of the number is passed to the data input D_{in} , i.e., D_3 . We will apply the clock, and this time the value of D_3 is 1. The first flip flop, i.e., FF-3, is set, and the word is stored in the register at the first falling edge of the clock. Now, the stored word is 1000.



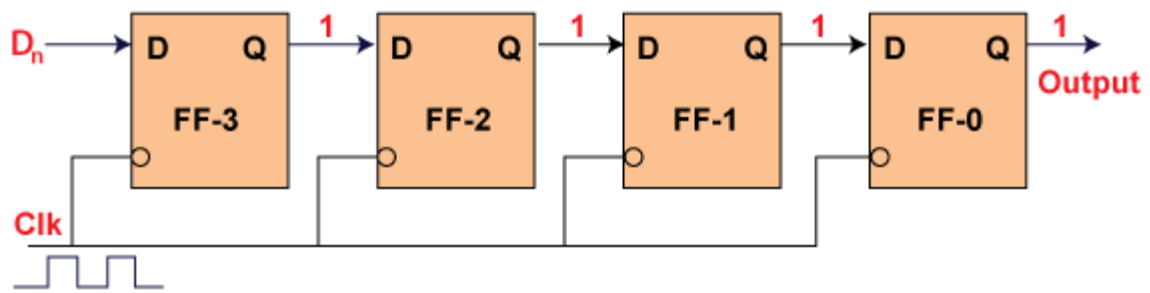
The next bit of the binary number, i.e., 1, is passed to the data input D_2 . The second flip flop, i.e., FF-2, is set, and the word is stored when the next negative edge of the clock hits. The stored word is changed to 1100.



The next bit of the binary number, i.e., 1, is passed to the data input D_1 , and the clock is applied. The third flip flop, i.e., FF-1, is set, and the word is stored when the negative edge of the clock hits again. The stored word is changed to 1110.



Similarly, the last bit of the binary number, i.e., 1, is passed to the data input D_0 , and the clock is applied. The last flip flop, i.e., FF-0, is set, and the word is stored when the clock's negative edge arrives. The stored word is changed to 1111.

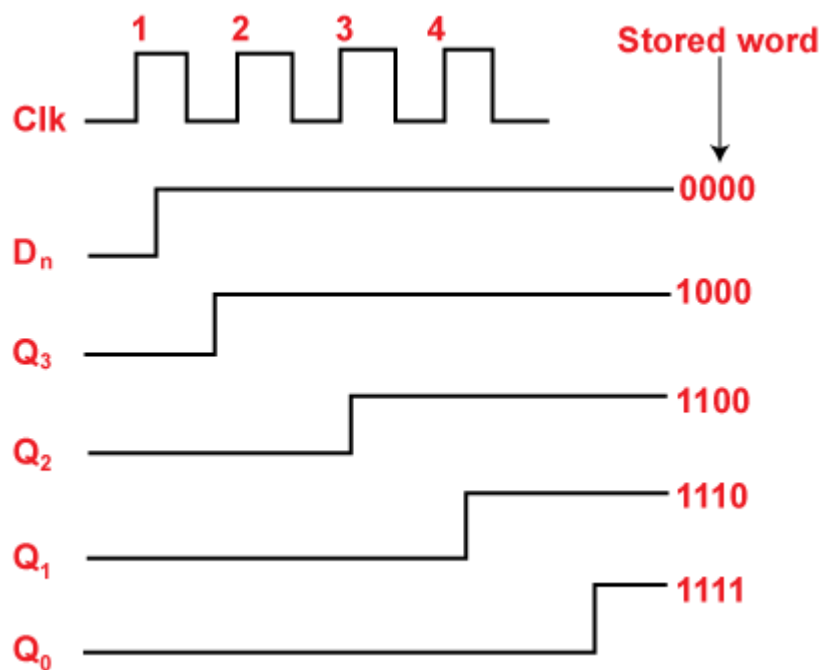


Truth Table

	Clk	$D_n = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(1)	↓	1 →	1	0	0	0
(2)	↓	1 →	1	1	0	0
(3)	↓	1 →	1	1	1	0
(4)	↓	1 →	1	1	1	1

→ Direction of data travel

Waveforms

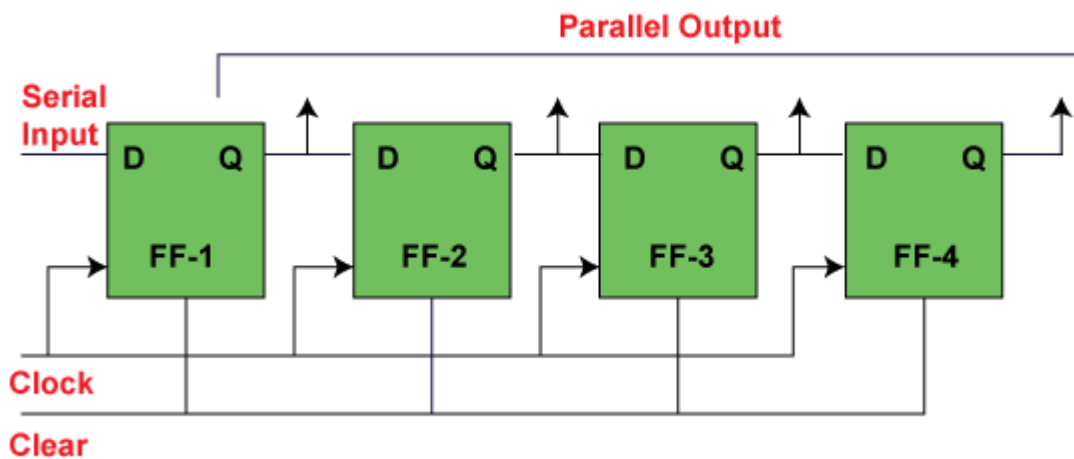


Serial IN Parallel OUT

In the "**Serial IN Parallel OUT**" shift register, the data is passed serially to the flip flop, and outputs are fetched in a parallel way. The data is passed bit by bit in the register, and the output remains disabled until the data is not passed to the data input. When the data is passed to the register, the outputs are enabled, and the flip flops contain their return value

Below is the block diagram of the 4-bit **serial in the parallel-out** shift register. The circuit having four D flip-flops contains a clear and clock signal to reset these four flip flops. In **SIPO**, the input of the second flip flop is the output of the first flip flop, and so on. The same clock signal is applied to each flip flop since the flip flops synchronize each other. The parallel outputs are used for communication.

Block Diagram



Parallel IN Serial OUT

In the "**Parallel IN Serial OUT**" register, the data is entered in a parallel way, and the outcome comes serially. A four-bit "**Parallel IN Serial OUT**" register is designed below. The input of the flip flop is the output of the previous Flip Flop. The input and outputs are connected through the combinational circuit. Through this combinational circuit, the binary input B_0 , B_1 , B_2 , B_3 are passed. The **shift mode** and the **load mode** are the two modes in which the "**PISO**" circuit works.

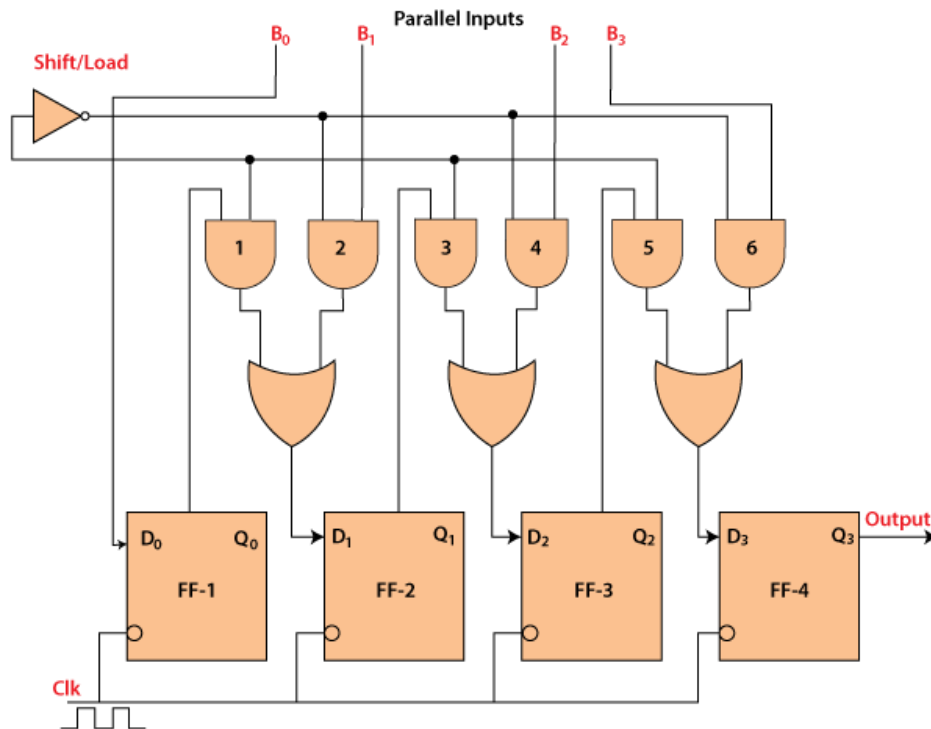
Load mode

The bits B_0 , B_1 , B_2 , and B_3 are passed to the corresponding flip flops when the second, fourth, and sixth "AND" gates are active. These gates are active when the shift or load bar line set to 0. The binary inputs B_0 , B_1 , B_2 , and B_3 will be loaded into the respective flip-flops when the edge of the clock is low. Thus, parallel loading occurs.

Shift mode

The second, fourth, and sixth gates are inactive when the load and shift line set to 0. So, we are not able to load data in a parallel way. At this time, the first, third, and fifth gates will be activated, and the shifting of the data will be left to the right bit. In this way, the **"Parallel IN Serial OUT"** operation occurs.

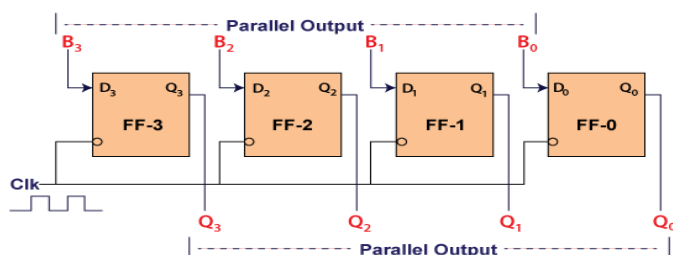
Block Diagram



Parallel IN Parallel OUT

In **"Parallel IN Parallel OUT"**, the inputs and the outputs come in a parallel way in the register. The inputs A₀, A₁, A₂, and A₃, are directly passed to the data inputs D₀, D₁, D₂, and D₃ of the respective flip flop. The bits of the binary input is loaded to the flip flops when the negative clock edge is applied. The clock pulse is required for loading all the bits. At the output side, the loaded bits appear.

Block Diagram



A special type of sequential circuit used to count the pulse is known as a counter, or a collection of flip flops where the clock signal is applied is known as counters.

The counter is one of the widest applications of the flip flop. Based on the clock pulse, the output of the counter contains a predefined state. The number of the pulse can be counted using the output of the counter.

Truth Table

Clock	Counter output		State number	Decimal counter output
	Q_B	Q_A		
Initially	0	0	-	0
1 st	0	1	1	1
2 nd	1	0	2	2
3 rd	1	1	3	3
4 th	0	0	4	0

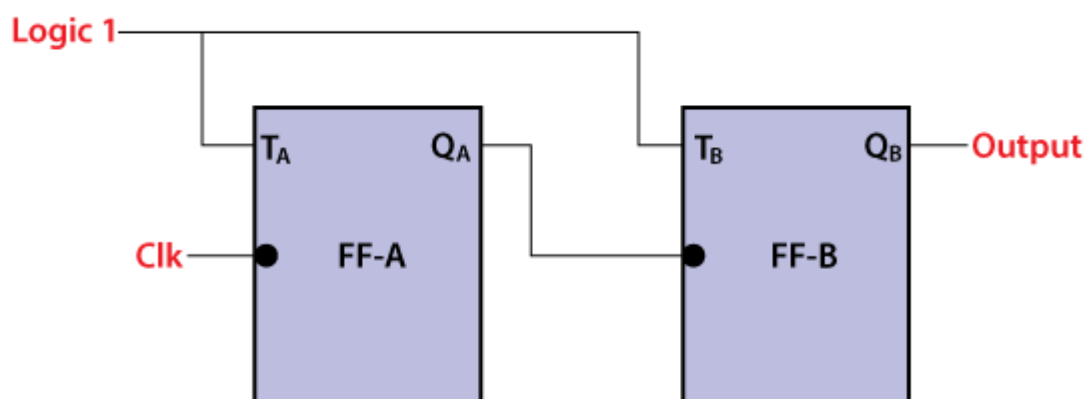
There are the following types of counters:

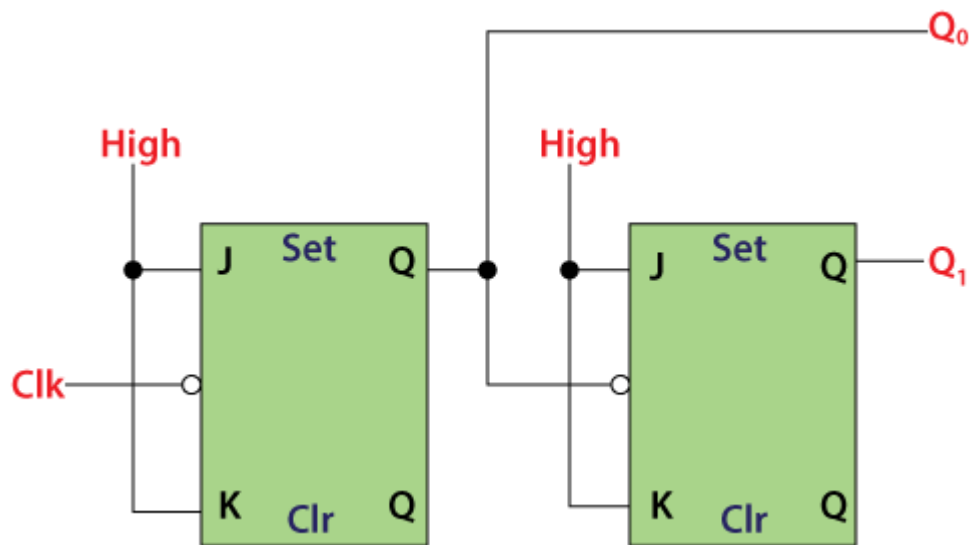
- Asynchronous Counters
- Synchronous Counters

Asynchronous or ripple counters

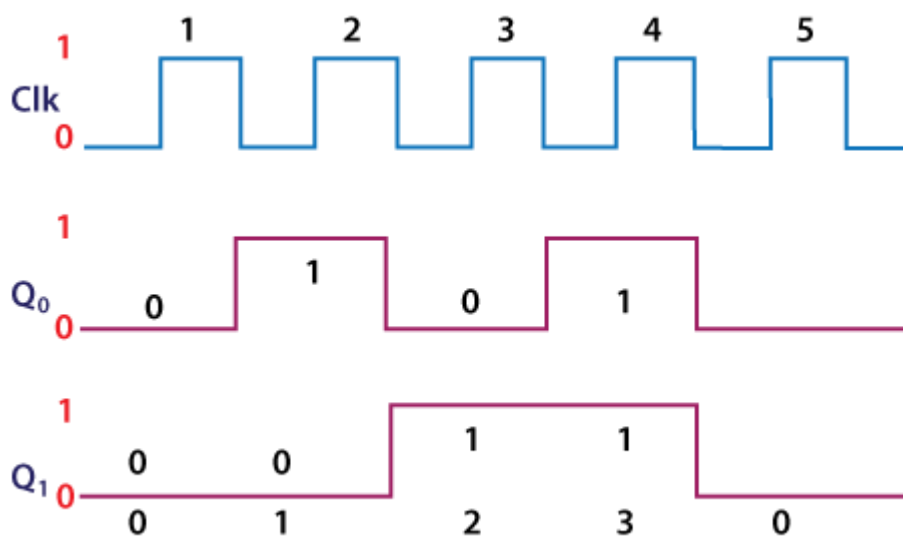
The **Asynchronous counter** is also known as the **ripple counter**. Below is a diagram of the 2-bit **Asynchronous counter** in which we used two T flip-flops. Apart from the T flip flop, we can also use the JK flip flop by setting both of the inputs to 1 permanently. The external clock pass to the clock input of the first flip flop, i.e., FF-A and its output, i.e., is passed to clock input of the next flip flop, i.e., FF-B.

Block Diagram





Signal Diagram



Operation

- Condition 1:** When both the flip flops are in reset condition.
Operation: The outputs of both flip flops, i.e., Q_A Q_B , will be 0.
- Condition 2:** When the first negative clock edge passes.
Operation: The first flip flop will toggle, and the output of this flip flop will change from 0 to 1. The output of this flip flop will be taken by the clock input of the next flip flop. This output will be taken as a positive edge clock by the second flip flop. This input will not change the second flip flop's output state because it is the negative edge triggered flip flop.
 So, $Q_A = 1$ and $Q_B = 0$
- Condition 3:** When the second negative clock edge is applied.
Operation: The first flip flop will toggle again, and the output of this flip flop will

change from 1 to 0. This output will be taken as a negative edge clock by the second flip flop. This input will change the second flip flop's output state because it is the negative edge triggered flip flop.

So, $Q_A = 0$ and $Q_B = 1$.

4. **Condition 4:** When the third negative clock edge is applied.

Operation: The first flip flop will toggle again, and the output of this flip flop will change from 0 to 1. This output will be taken as a positive edge clock by the second flip flop. This input will not change the second flip flop's output state because it is the negative edge triggered flip flop.

So, $Q_A = 1$ and $Q_B = 1$

5. **Condition 5:** When the fourth negative clock edge is applied.

Operation: The first flip flop will toggle again, and the output of this flip flop will change from 1 to 0. This output will be taken as a negative edge clock by the second flip flop. This input will change the output state of the second flip flop.

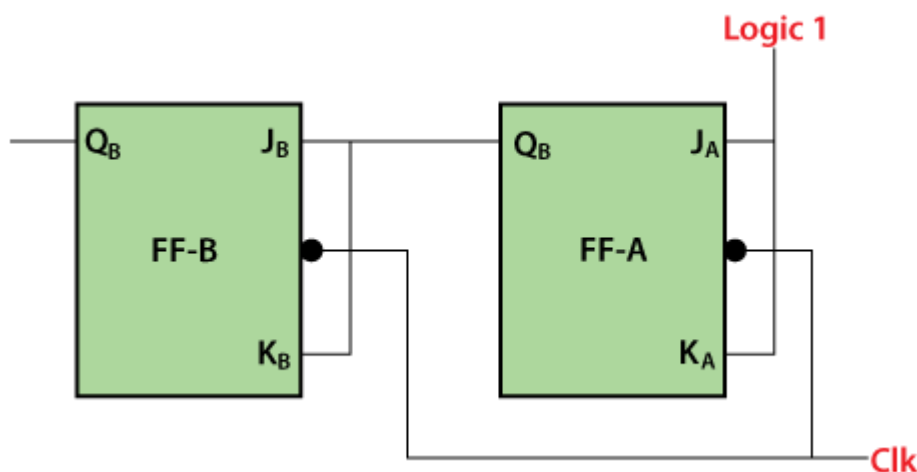
So, $Q_A = 0$ and $Q_B = 0$

Synchronous counters

In the **Asynchronous counter**, the present counter's output passes to the input of the next counter. So, the counters are connected like a chain. The drawback of this system is that it creates the counting delay, and the propagation delay also occurs during the counting stage. The **synchronous counter** is designed to remove this drawback.

In the **synchronous counter**, the same clock pulse is passed to the clock input of all the flip flops. The clock signals produced by all the flip flops are the same as each other. Below is the diagram of a 2-bit synchronous counter in which the inputs of the first flip flop, i.e., FF-A, are set to 1. So, the first flip flop will work as a toggle flip-flop. The output of the first flip flop is passed to both the inputs of the next JK flip flop.

Logical Diagram



of the first flip flop will be 1. The clock input of the first flip flop and both of its inputs will set to 1. In this way, the state of the second flip flop will change from 0 to 1.

So, $Q_A = 0$ and $Q_B = 1$

4. **Condition 2:** When the third negative clock edge passes.

Operation: The first flip flop will toggle from 0 to 1, but at this instance, both the inputs and the clock input set to 0. Hence, the outputs will remain the same as before.

So, $Q_A = 1$ and $Q_B = 1$

5. **Condition 2:** When the fourth negative clock edge passes.

Operation: The first flip flop will toggle from 1 to 0. At this instance, the inputs and the clock input of the second flip flop set to 1. Hence, the outputs will change from 1 to 0.

So, $Q_A = 0$ and $Q_B = 0$

Ripple Counter

Ripple counter is a special type of **Asynchronous** counter in which the clock pulse ripples through the circuit. The n-MOD ripple counter forms by combining n number of flip-flops. The n-MOD ripple counter can count 2^n states, and then the counter resets to its initial value.

Features of the Ripple Counter:

- Different types of flip flops with different clock pulse are used.
- It is an example of an asynchronous counter.
- The flip flops are used in toggle mode.
- The external clock pulse is applied to only one flip flop. The output of this flip flop is treated as a clock pulse for the next flip flop.
- In counting sequence, the flip flop in which external clock pulse is passed, act as LSB.

Based on their circuitry design, the counters are classified into the following types:

Up Counter

The up-counter counts the states in ascending order.

Down Counter

The down counter counts the states in descending order.

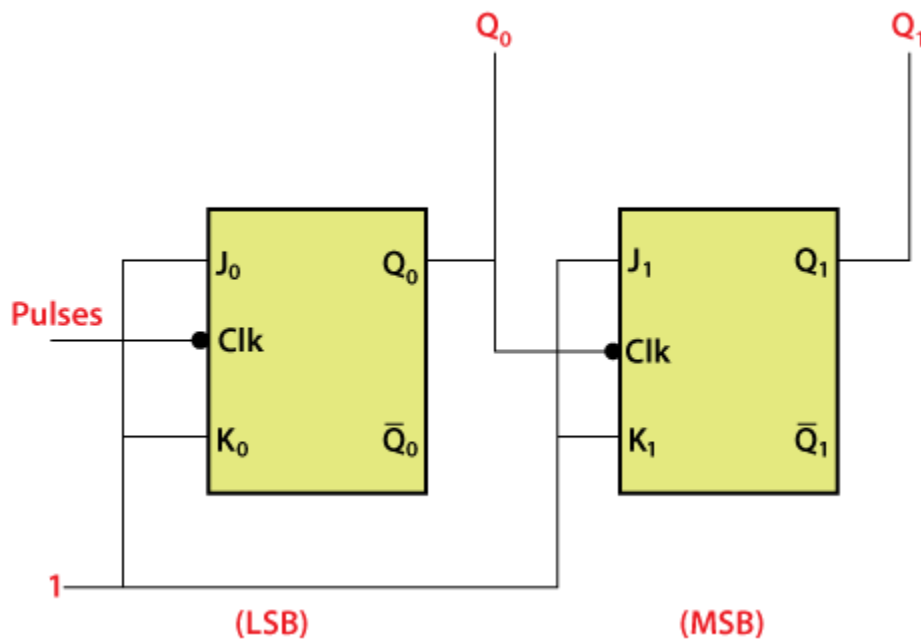
Up-Down Counter

The up and down counter is a special type of bi-directional counter which counts the states either in the forward direction or reverse direction. It also refers to a reversible counter.

Binary Ripple Counter

A **Binary counter** is a **2-Mod counter** which counts up to 2-bit state values, i.e., $2^2 = 4$ values. The flip flops having similar conditions for toggling like T and JK are used to construct the **Ripple counter**. Below is a circuit diagram of a **binary ripple counter**.

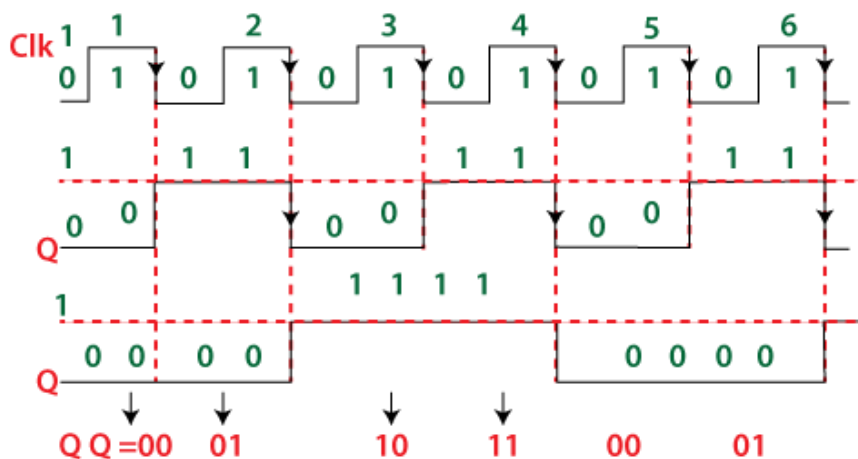
In the circuit design of the binary ripple counter, two JK flip flops are used. The high voltage signal is passed to the inputs of both flip flops. This high voltage input maintains the flip flops at a state 1. In JK flip flops, the negative triggered clock pulse use.



The outputs Q_0 and Q_1 are the LSB and MSB bits, respectively. The truth table of JK flip flop helps us to understand the functioning of the counter.

J_n	K_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	\bar{Q}_n

When the high voltage to the inputs of the flip flops, the fourth condition is of the JK flip flop occurs. The flip flops will be at the state 1 when we apply high voltage to the input of the flip-flop. So, the states of the flip flops passes are toggled at the negative going end of the clock pulse. In simple words, the flip flop toggle when the clock pulse transition takes place from 1 to 0.



The state of the output Q_0 change when the negative clock edge passes to the flip flop. Initially, all the flip flops are set to 0. These flip flop changes their states when the passed clock goes from 1 to 0. The JK flip flop toggles when the inputs of the flip flops are one, and then the flip flop changes its state from 0 to 1. For all the clock pulse, the process remains the same.

Number of input pulses	Q_1	Q_0
0	-	-
1	0	0
2	0	1
3	1	0
4	1	1

The output of the first flip flop passes to the second flip flop as a clock pulse. From the above timing diagram, it is clear that the state of the second flip flop is changed when the output Q_0 goes transition from 1 to 0. The outputs Q_0 and Q_1 treat as LSB and MSB. The counter counts the values 00, 01, 10, 11. After counting these values, the counter resets itself and starts counting again from 00, 01, 10, and 1. The count values until the clock pulses are passed to J_0K_0 flip flop.

Ring Counter

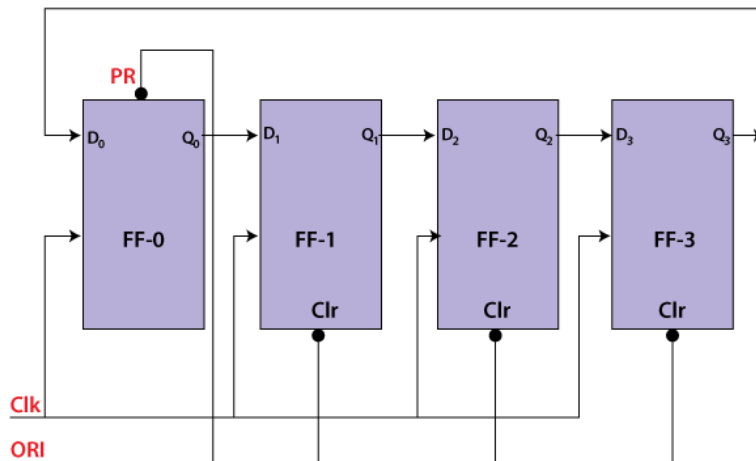
A **ring counter** is a special type of application of the **Serial IN Serial OUT** Shift register. The only difference between the shift register and the ring counter is that the last flip flop outcome is taken as the output in the shift register. But in the ring counter, this outcome is passed to the first flip flop as an input. All of the remaining things in the ring counter are the same as the shift register.

In the **Ring counter**

No. of states in Ring counter = No. of flip-flop used

Below is the block diagram of the 4-bit ring counter. Here, we use 4 **D flip flops**. The same clock pulse is passed to the clock input of all the flip flops as a synchronous counter. The **Overriding input (ORI)** is used to design this circuit.

The Overriding input is used as **clear** and **pre-set**.



The output is 1 when the pre-set set to 0. The output is 0 when the clear set to 0. Both PR and CLR always work in value 0 because they are active low signals.

1. PR = 0, Q = 1
2. CLR = 0, Q = 0

These two values (always fixed) are independent with the input D and the Clock pulse (CLK).

Working

The ORI input is passed to the PR input of the first flip flop, i.e., FF-0, and it is also passed to the clear input of the remaining three flip flops, i.e., FF-1, FF-2, and FF-3. The pre-set input set to 0 for the first flip flop. So, the output of the first flip flop is one, and the outputs of the remaining flip flops are 0. The output of the first flip flop is used to form the ring in the **ring counter** and referred to as **Pre-set 1**.

ORI	Clk	Q ₀	Q ₁	Q ₂	Q ₃
low	X	1	0	0	0
high	low	0	1	0	0
high	low	0	0	1	0
high	low	0	0	0	1
high	low	1	0	0	0

In the above table, the highlighted 1's are **pre-set 1**.

The **Pre-set 1** is generated when

- ORI input set to low, and that time the Clk doesn't care.
-
- When the ORI input set to high and the low clock pulse signal is passed as the negative clock edge triggered.

A ring forms when the **pre-set 1** is shifted to the next flip-flop at each clock pulse.

So, 4-bit counter, 4 states are possible which are as follows:

1. 1 0 0 0
2. 0 1 0 0
3. 0 0 1 0
4. 0 0 0 1

Answer the following questions:

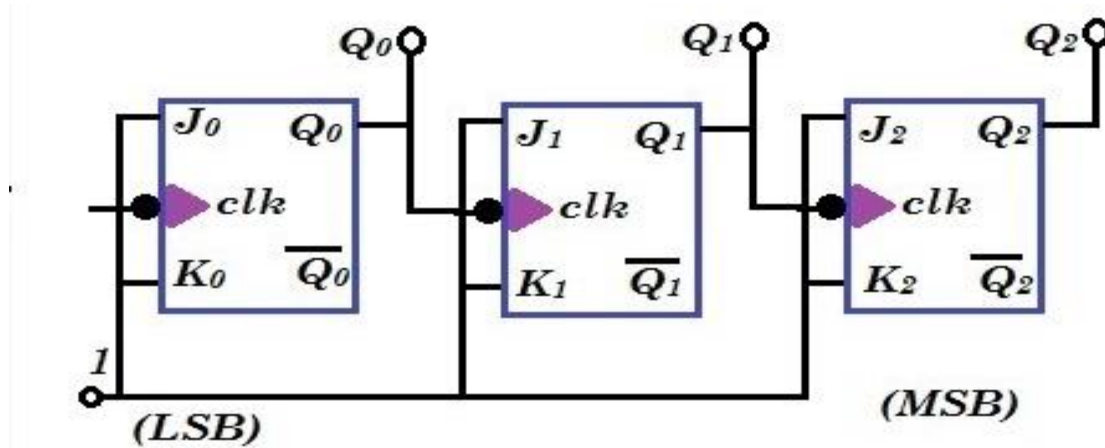
1. What is the difference between synchronous and asynchronous counter?

Ans:

Synchronous Counter	Asynchronous Counter
In synchronous counter, all flip flops are triggered with same clock simultaneously.	In asynchronous counter, different flip flops are triggered with different clock, not simultaneously.
Synchronous Counter is faster than asynchronous counter in operation.	Asynchronous Counter is slower than synchronous counter in operation.
Synchronous Counter does not produce any decoding errors.	Asynchronous Counter produces decoding error.
Synchronous Counter is also called Parallel Counter.	Asynchronous Counter is also called Serial Counter.
Synchronous Counter designing as well implementation is complex due to increasing the number of states.	Asynchronous Counter designing as well as implementation is very easy.
Synchronous Counter will operate in any desired count sequence.	Asynchronous Counter will operate only in fixed count sequence (UP/DOWN).
Synchronous Counter examples are: Ring counter, Johnson counter.	Asynchronous Counter examples are: Ripple UP counter, Ripple DOWN counter.
In synchronous counter, propagation delay is less.	In asynchronous counter, there is high propagation delay.

2. What do you understand by ripple (asynchronous) counter? Draw the circuit for 3 bit ripple counter using J-K flip-flop?

Ans: A ripple counter is an asynchronous counter where only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop. Asynchronous counters are also called ripple-counters because of the way the clock pulse ripples its way through the flip-flops.

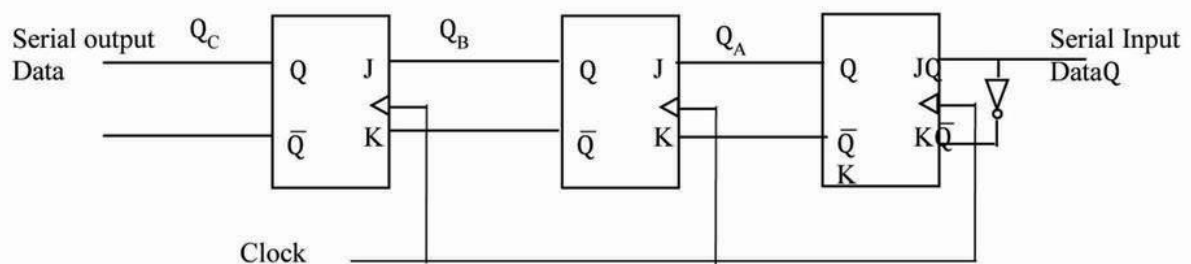


3. What is modulus counter?

Ans: Modulus Counters, or simply *MOD counters*, are defined based on the number of states that the counter will sequence through before returning back to its original value. For example, a 2-bit counter that counts from 00_2 to 11_2 in binary, that is 0 to 3 in decimal, has a modulus value of 4 ($00 \rightarrow 1 \rightarrow 10 \rightarrow 11$, and return back to 00) so would therefore be called a modulo-4, or mod-4, counter.

4. Design a SISO left shift register using J-K flip-flops.

Ans:



5. What is shift register? What are its various mode of operation?

Ans: A group of flip flops which is used to store multiple bits of data and the data is moved from one flip flop to another is known as **Shift Register**. The bits stored in registers shifted when the clock pulse is applied within and inside or outside the registers. To form an n-bit shift register, we have to connect n number of flip flops. So, the number of bits of the binary number is directly proportional to the number of flip flops. The flip flops are connected in such a way that the first flip flop's output becomes the input of the other flip flop.

A **Shift Register** can shift the bits either to the left or to the right. A **Shift Register**, which shifts the bit to the left, is known as "**Shift left register**", and it shifts the bit to the right, known as "**Right left register**".

6. Mention some of the features of ripple counters.

Ans:

- It is an asynchronous counter.
- Different flip-flops are used with a different clock pulse.
- All the flip-flops are used in toggle mode.
- Only one flip-flop is applied with an external clock pulse and another flip-flop clock is obtained from the output of the previous flip-flop.
- The flip-flop applied with external clock pulse act as LSB (Least Significant Bit) in the counting sequence.

7. List application of counters.

Ans: Some of their applications are listed below.

- Frequency counters
- Digital clocks
- Analog to digital convertors.
- With some changes in their design, counters can be used as frequency divider circuits. The frequency divider circuit is that which divides the input frequency exactly by '2'.
- In time measurement. That means calculating time in timers such as electronic devices like ovens and washing machines.
- We can design digital triangular wave generator by using counters.

8. What do you mean by ring counter?

Ans: Ring counter is a typical application of Shift register. Ring counter is almost same as the shift counter. The only change is that the output of the last flip-flop is connected to the input of the first flip-flop in case of ring counter but in case of shift register it is taken as output.